Handbook Formulas for Predictive Analysis

Gianmaria Pizzo - 872966

2022/2023

Contents

1 Disclaimer

This handbook is intended to be a quick recap for the course, it should be use as a way to revise for the final exam and does not substitute the professor's slides/notes nor the textbooks. The content here is thought for students who have difficulties with recognizing the meaning of the variables or have troubles understanding how to pass from the theoretical level to practical applications. Please consult your professor and ask them to check this material before using it, as it might contain errors.

2 Notation for SLR

Let us start by giving some context for the simple linear regression model using the least squares method.

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Y, *target* or *response* variable. It is a random variable;
  - $\widehat{Y}$, the estimated model
    - `fit <- lm(formula = target ~ predictor, data = dataset)`[1]
  - y, vector of **observed** (real) values for the response variable;
    - `y <- dataset$target`
  - y_i, the i-th observed value;
    - `y[i]` with $i = 1$, to n
  - $\widehat{y}$, vector of **estimated** values for the response variable;
    - `y_hat <- fitted(fit)`
  - $\widehat{y}_i$, the i-th estimated value;
    - `y_hat[i]` with $i = 1$, to n
- X, *input* or *predictor* variable. It is a random variable;
  - x vector of **observed** (real) values for the input variable;
    - `x <- dataset$predictor`
  - x_i, the i-th predictor
    - `x[i]` with $i = 1$, to n
- β_0 (the intercept) and β_1 (the angular coefficient) are the beta parameters, which model our regression.
  - `beta0_hat <- fit$coefficients[1]`
  - `beta1_hat <- fit$coefficients[2]`
- ε is the noise variable, but we will focus on the residuals e_i
  - `residuals <- fit$residuals`
  - `residuals <- residuals(fit)`
- n, number of observations
  - `n <- rows(dataset)`
- $p - 1$ number of predictors (from x_1 to x_p)
- p number of beta-parameters (from β_0 to β_{p-1})
  - `p <- 2` for this case

[1] Check the possible callables and operations through names(fit)

3 SLR - Least Squares Method

The goal here is to predict $Y|X = x_i$, by taking a linear function $Y = m(X) = \beta_0 + \beta_1 X$ and finding the parameters (β_0*, β_1*) that minimize the MSE, which assesses the goodness of our prediction. Since, we are dealing with samples we use estimates and hope the **law of large numbers** will help us converge to the best result. Our estimated model will be:

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X + \varepsilon$$

3.1 Assumptions for the model

The simple linear regression model is a statistical model for two variables, X and Y . We use X the predictor variable to try to predict Y , the target or response. The assumptions of the model are:

- The distribution of X is **arbitrary** (and perhaps X is even non-random).
  - No assumptions about the marginal distributions of the variables or about the joint distribution of the two variables. No assumptions about the fluctuations of Y around the optimal regression line. No assumption that X came before Y in time, or that X causes Y.
- $Y = \beta_0 + \beta_1 X + \varepsilon$, for some coefficients (the beta parameters) and some noise variable ε
  - The assumption of a **linear functional form** for the relationship between Y and X, is non trivial. In fact, they could have quadratic relationship.
  - The assumption of **additive noise** is non-trivial. Its not absurd to imagine that either measurement error or fluctuations might change Y multiplicatively (for instance).
- The noise variable may represent measurement error, fluctuations in Y, the effect of the variables which affect Y which we have not (can not) include in the model.
  - $\mathbb{E}[\varepsilon|X = x] = 0$, no matter what x is
    - the noise variables ε_i all have the same expectation (0).
  - $Var[\varepsilon|X = x] = \sigma^2$, constant variance or **homoskedasticity** assumption is non-trivial
    - the noise variables ε_i all have the same variance (σ^2)
  - $Cov[\varepsilon_i, \varepsilon_j] = 0$, (unless $i = j$ of course) non-correlation assumption is non-trivial
  - ε is uncorrelated across elements

3.2 Interpretation

Some of the following formulas can be derived from the estimating equations only because we are considering a linear model:

- The intercept is estimated by $\hat{\beta}_0$, which dictates where the line passes through. It is unbiased.
  - This forces the regression to pass through $(\bar{x}, \bar{y})$;
  - It should have the same units of Y.
- The slope is estimated by $\hat{\beta}_1$, which dictates how strong the relation between X and Y is. It is unbiased.
  - It is explains how much Y grows when X grows of a unit of measure (Y's unit of measure/ X's unit of measure). For example $\$/hour$, the angular coefficient explains how much more dollars we get for one more hour of work.
  - The regression slope is a re-weighting of the Pearsons correlation coefficient which accounts for the variability of the individual variables. The sign of $\hat{\beta}_1$ is the sign of r_{xy}. It increases the more the X and Y tend to fluctuate together.
    - If positive the slope increases upwards as we move to the right, this implies Y and X are strongly related. Else, it decreases.
    - However, when really strong and positive it may be a sign that it is not working well. In fact, this might mean that two variables are not linearly related.
    - If it is 0, we are basically estimating the mean and means the model is no good.

  - We want its variance to be as small as possible, as more variability means there could be too many values to choose from:
    * σ^2 should be small.
    * n should be as large as possible, less uncertainty.
    * s_x^2 should be $]0, \inf[$, and large.

- It can be shown that the least square estimates are in some sense optimal: they are the minimum-variance mean-unbiased estimators for the simple linear model (when the assumptions are valid)

- The errors do not coincide with the residuals of the models $\varepsilon_i \neq e_i$, although they share a relationship that is "good enough" for us to use them as way to check our model. Let us see some characteristics of the empirical residuals:
  - $\frac{1}{n}\sum_{i=1}^{n} e_i(x_i - \bar{x}) = 0$, The residuals are always uncorrelated/independent of the x_i. They could show non linear dependence with the predictor which is why we would need a more complex model. When plotting **Residuals vs Predictor** we want to be sure there is no pattern (it should appear like sparse dots in the graph) or we are likely to be forgetting some other important relations. In this case we can plot **residuals against another predictor** or against a **transformation of the variable**.
    * `cor(residuals(fit), dataset$predictor)`
  - $\bar{e_i} = 0$, by construction and they **should have expectation zero** conditional on x, $\mathbb{E}[e_i|X = x] = 0$
    * `mean(residuals(fit))` should be null
  - $Var[e_i] = \sigma^2$, **constant variance**, unchanging with x. This is something we need to check.
    * `summary(fit)$sigma` is s_e
  - The residuals cant be completely **uncorrelated with each other**, but the correlation should be extremely weak, and grow negligible as n approaches infinite.

In the end, each of these points leads to a diagnostic and need to be checked in a model.

3.3 Measures of model checking

3.3.1 Decomposition of the sum of squares

Again, goal is to minimize the errors, which we can do by decomposing the total variance. This helps us understand how much variance of Y is explained by the model vs how much variance is still to be included.

$$SS_{TOT} = SS_{REG} + SS_{RES}$$

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y_i} - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

We focus on the SS_{RES} which increases when $\hat{y}$ is very far from the mean, and decreases when there is not a relevant relationship between X and Y.

3.3.2 R^2

The coefficient of determination R^2 is the proportion of variability in the response that is accounted for by the statistical model. It only judges the goodness of the model's adaptability.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{SS_{REG}}{SS_{TOT}} \in [0, 1]$$

- `r2 <- summary(fit)$r.squared`

- We say "$(R^2) \cdot 100$ percent of Y's variability is explained by X". In the case of the MLR this represents "$(R^2) \cdot 100$ percent of the observed variability in the target is explained by the linear relationship with the **p** predictors". It can be found in the summary, under the `Multiple R-squared` name.

- However it says nothing regarding quality of the model. In fact, **it favors more complex models, with multiple estimated parameters since their presence captures more variability** but puts the model at a high risk of overfitting. This is why we always need to plot the data and the residuals (against the predictor X).

- The variability increases when the noise around the regression line σ^2: the noisier the data the more variable the estimated regression line, and the more of that noise will propagate into our predictions. It decreases as we have more observations (n), which are further spread out along the horizontal axis (s^2).

4 Formulas for Least Squares Estimates

All the estimators used, are the results of the assumptions we made at the start.

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta_1} X + \varepsilon$$

```
fit <-lm(formula = target ~ predictor, data = dataset)
```

- $\mathbb{E}[\widehat{Y}|x_1, ..., x_n] = \beta_0 + \beta_1 x$
- $Var[\widehat{Y}|x_1, ..., x_n] = \frac{s_e^2}{n} + \left(1 + \frac{(x-\bar{x})^2}{s_x^2}\right) = \left(\frac{SSE}{n-2} \cdot \frac{1}{n}\right) + \left(1 + \frac{(x-\bar{x})^2}{s_x^2}\right)$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1$$

It can be found like this:

```
beta0_hat <- y_bar - beta1_hat * x_bar
beta0_hat <- se * sqrt(1/n + mean(dataset$predictor)^2/(n * s2x))
```

Or it can be shown under the `Estimate` column in the summary of the model

- $\mathbb{E}[\widehat{\beta_0}|x_1, ..., x_n] = \beta_0$, unbiased
- $Var[\widehat{\beta_0}|x_1, ..., x_n] = s_e^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{ns_X^2}\right] = \frac{SSE}{n-2} \cdot \left[\frac{1}{n} + \frac{\bar{x}^2}{ns_X^2}\right]$
- $SE[\widehat{\beta_0}] = s_e \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{ns_x^2}} = \sqrt{\frac{SSE}{n-2} \cdot (\frac{1}{n} + \frac{\bar{x}^2}{ns_x^2})}$ which appears under the `Std. Error` column in the summary of the model

---

$$\hat{\beta}_1 = \frac{c_{xy}}{s^2 x} = r_{xy} \cdot \frac{s_y}{s_x}$$

It can be found like this:

```
beta1_hat <- rxy * sqrt(s2y/s2x)
beta1_hat <-  se * sqrt(1/(n * s2x))
```

Or it can be shown under the `Estimate` column in the summary of the model

- $\mathbb{E}[\widehat{\beta_1}|x_1, ..., x_n] = \beta_1$, unbiased
- $Var[\widehat{\beta_1}|x_1, ..., x_n] = \frac{s_e^2}{ns_x^2} = \frac{SSE}{n-2} \cdot \frac{1}{ns_x^2}$
- $SE[\widehat{\beta_1}] = \frac{s_e}{\sqrt{ns_x^2}} = \sqrt{\frac{SSE}{n-2} \cdot \frac{1}{ns_x^2}}$ which appears under the `Std. Error` column in the summary of the model

Base sample formulas:

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, the sample mean
  - `x_bar <- mean(x)`
- $s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$, the sample variance
  - $s_x^2 > 0$ for $\hat{\beta}_1$ to be defined: if there is no variation in X we can not really say much about the effect that X has on Y
  - `s2x <- sum((x-x_bar)^2)/n`
- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, the sample mean
  - `y_bar <- mean(y)`
- $s_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$, the sample variance
  - `s2y <- sum((y - y_bar)^2)/n`
- $c_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n}(\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y})$, the sample covariance
  - `covxy <- cov(x,y)`
- $r_{xy} = \frac{c_{xy}}{s_x s_y}$
  - `rxy <- cor(x,y)`
- $e_i = y_i - \widehat{m}(x_i) = y_i - (\widehat{\beta}_0 + \widehat{\beta_1 x_i})$, the residuals of the model
  - We use e_i for our model-checking and diagnostics instead of ε_i, since they share an important relationship
  - `e_i <- y - y_hat`
  - `e_i <- residuals(fit)`
- $SSE = \sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2 = \sum_{i=1}^{n}(y_i - (\widehat{\beta}_0 + \widehat{\beta_1 x_i}))^2$ the sum of squared errors, which appears as the `Residual standard error` on $n - p$ degrees of freedom in R.
  - `SSE <- sum((e_i)^2)`
  - `SSE <- sum((y - y_hat)^2)`
  - `SSE <- sum((residuals(fit))^2`
- $s_e^2 = \frac{n}{n-2} \cdot \widehat{\sigma}^2 = \frac{n}{n-2} \cdot \frac{1}{n} SSE = \frac{SSE}{n-2} = \frac{\sum_{i=1}^{n}(y_i - (\widehat{\beta}_0 + \widehat{\beta_1 x_i}))^2}{n-2}$
  - The minimal value of the in-sample MSE for this model, is a natural estimator for σ^2. However, we use this estimator since it is unbiased.
  - `se <- sqrt(SSE/(n-2))`
  - `se <- summary(fit)$sigma`
  - `se <- sqrt(sum((dataset$target - fitted(fit))^2)/(nrow(dataset)-2))`

5 SLR with Gaussian Noise - Maximum Likelihood Method

In the method of maximum likelihood, we pick the parameter values which maximize the likelihood, or, equivalently, maximize the log-likelihood.

5.1 New assumptions

If we made more detailed assumptions about the distribution of ε, we could make more precise inference:

- The most common choice is to assume ε follows a Gaussian/Normal distribution.
  - $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- ε is uncorrelated across observations

Under these conditions, which are added to the previous ones, we can transform the model to make it behave like this:

- $[Y|X = x] \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2)$
- Y_i, Y_j are independent give X_i, X_j

Given any data set $(x1, y1), (x2, y2), ...(xn, yn)$, we can now write down the probability density, under the model, of seeing that data through the log-likelihood. We can then maximize it through the usual method of getting the derivatives for each beta parameter and setting them to 0. It is evident by now that, the least squares solution is the Maximum Likelihood Estimate and the SLR with Gaussian Noise Model!

What makes the Gaussian noise assumption important is that it gives us an exact conditional distribution for each Y_i, and this in turn gives us a distribution the sampling distribution for the estimators. Remember that we can write the estimates for the beta parameters in the form "constant plus sum of noise variables".

Suppose that Y_i, $i = 1, ..., n$ are n independent random variables with $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and w_i, $i = 0, ..., n$ are real constants. Then:

$$w_0 + \sum_{i=1}^{n} w_i Y_i \sim \mathcal{N}(w_0 + \sum_{i=1}^{n} w_i \mu_i, \sum_{i=1}^{n} w_i^2 \sigma_i^2)$$

6 Formulas for Maximum Likelihood Method

Using the usual estimator s_e^2 for the variance we derive the following sampling distributions:

$$\widehat{m}(x) \sim \mathcal{N}\left(\beta_0 + \beta_1 x, \sigma^2\left(\frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right)\right)$$

$$\downarrow$$

$$\frac{\widehat{m}(x) - Y}{SE[\widehat{m}(x)]} \sim t_{n-2}$$

$$\downarrow$$

$$SE[\widehat{m}(x)] = s_e\sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

$$\widehat{\beta}_0 = \sum_{i=1}^n w_i^* Y_i = \bar{Y} - \widehat{\beta}_1 \bar{x} \sim \mathcal{N}\left(\beta_0, \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right)\right)$$

$$\downarrow$$

$$\frac{\widehat{\beta}_0 - \beta_0}{SE[\widehat{\beta}_0]} \sim t_{n-2}$$

$$\downarrow$$

$$SE[\widehat{\beta}_0] = s_e\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

$$\widehat{\beta}_1 = \sum_{i=1}^n w_i Y_i = \frac{\sum_{i=1}^n (x_i - \bar{x}) Y_i}{\sum_{i=1}^n (x_i - \bar{x})^2} \sim \mathcal{N}\left(\beta_1, \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right)$$

$$\downarrow$$

$$\frac{\widehat{\beta}_1 - \beta_1}{SE[\widehat{\beta}_1]} \sim t_{n-2}$$

$$\downarrow$$

$$SE[\widehat{\beta}_1] = s_e \frac{1}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

$$s_e^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$$

7 Inference for SLR

7.1 Confidence Intervals for $\widehat{\beta}_0, \widehat{\beta}_1$

$$est \pm CRIT \times SE$$

In R, a two tailed test for our fit, would be `confint(fit, level = alpha)`

- **est**, the estimate for parameters of interest
- **CRIT**, critical value derived from the distribution
  - `qt(c(0.0+(alpha/2), 1-(alpha/2)), df = n-p)`
  - For the t-student distribution $t_{\alpha/2,n-2}$ is the critical value such that $Pr(T_{n-2} > t_{\alpha/2,n-2}) = \alpha/2$
  - $\alpha/2$ determines the amplitude of the interval. The smaller alpha is, the larger the intervals.
  - $n - p$, the degrees of freedom. They take into account how many parameters are estimated.
- **SE**, standard error of the estimate

For our case:

$$\widehat{\beta}_0 \pm t_{\alpha/2,n-2} \times SE[\widehat{\beta}_0] \qquad \widehat{\beta}_1 \pm t_{\alpha/2,n-2} \times SE[\widehat{\beta}_1]$$

For example for a test:

```
    fit<- lm(dist ~ speed, data = cars)
    coef(fit)

[1] (Intercept) speed
    -17.579095  3.932409


    confint(fit, level = 0.99)

[1]             0.5 %       99.5 %
    (Intercept) -35.706610  0.5484205
    speed       2.817919    5.0468988
```

We can interpret it as, "we are 99 percent confident that if the difference in speed of a car and another is an increase of 1 mile per hour in the travelling speed, the average increase in stopping distance is between 2.818 and 5.047 feet, which is the interval for β_1"

Manually we can get the same result like this:

```
    # If we have only  the estimated values (est),
    # and the vcov() matrix, we want to test it like this:
    # we assume a significance level of 5%, alpha = 0.05
    est + qt(c(0.025, 0.975), df = n-2) * sqrt(vcov[i,j])
```

This will return two boundaries for the interval of parameter's estimated value: any value inside that range, is a value for which a null hypothesis **cannot be rejected**.

7.2 Confidence Intervals for $\widehat{m}(x)$

The uncertainty about estimate of the expected values of y, namely $\widehat{m}(x)$, is assessed through confidence intervals.

$$\widehat{m}(x) \pm t_{\alpha/2,n-2} \times SE[\widehat{m}(x)]$$

While `fitted()` returns the estimated values $\hat{y}_i$ for the observed sample predictors, to find confidence intervals for the mean response using R, we use the `predict()` function.

```
# S3 method for lm
predict(object, newdata, se.fit = FALSE,
scale = NULL, df = Inf,
interval = c("none", "confidence", "prediction"),
level = 0.95, type = c("response", "terms"),
terms = NULL, na.action = na.pass,
pred.var = res.var/weights, weights = 1,...)
```

We give the function our fitted model as well as new data, stored as a **data frame** where there should be at least one column with the same name as our predictor. (This is important, so that R knows the name of the predictor variable.)

For confidence intervals, the test in R would be more or less like the following:

```
predict(fit, newdata = new_dataset, interval = "confidence", level = alpha)
```

- Mind that `predict()` and `fitted()` return the same result for this kinds of models, but in MLR it is not the case.

7.3 Prediction Interval for new observations

The uncertainty about estimate of the expected values of y, namely $\widehat{m}(x)$, is assessed through predictive intervals. Here the variability of the error, plays a big role.

Sometimes we would like to predict a new observation, for a new value of x. We already have the best prediction for a new observation, namely $m(x)$. However, our estimate is still $\widehat{m}(x)$. What counts now is the distance, the difference in the amount of variability, $SE[\widehat{m}(x)]$. How far are our predictions from the real possible values?

We want to be able to interpolate and extrapolate, namely computing $\hat{y}$, **given potential values of x**. We can build confidence intervals for our predictions by exploiting our previous assumptions:

$$Y - \widehat{m}(x)|x_1, ..., x_n \sim \mathcal{N}\left(0, \sigma^2\left(1 + \frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)\right)$$

$$\downarrow$$

$$\widehat{m}(x) \pm t_{\alpha/2,n-2} \times se\sqrt{1 + \frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

To calculate this for a set of points in R notice there is only a minor change in syntax from finding a confidence interval for $m(x)$:

```
predict(fit, newdata = new_dataset, interval = "prediction", level = 0.99)
```

7.4 Evaluating uncertainty for predictions

To include the estimate of the uncertainty, we can use `se.fit = TRUE`. R will return

- `$fit`, the fitted values
- `$se.fit`, their s_e
- `$df`, the degrees of freedom
- `$residual.scale`, the s_e used from the model fit.

For example:

```r
# Model for dataset penguins
# fit's formula is flipper_length_mm ~ body_mass_g

# Prediction error for a penguin with
# body_mass_g = 5000 at alpha=0.5
pred2 <- predict(fit, newdata = data.frame(body_mass_g = 5000), se.fit = TRUE)
[1] 0.5269645

# We can access uncertainty through
predict(...)$se.fit
```

For better understanding we can plot the data to see where our predictions for the hypothesized values are positioned:

```r
# Model for dataset penguins
# fit's formula is flipper_length_mm ~ body_mass_g
plot(flipper_length_mm ~ body_mass_g,
    data = pengs, pch = 16, col = "grey60",
    xlim = c(2000, 9000), ylim = range(pred2$fit))

points(nd$body_mass_g, pred2$fit, col = "dodgerblue",
    pch = 4, cex = 1.9, lwd = 3)

abline(coef(fit), col = 2)
```

7.5 Evaluating the meaning of the intercept and angular coefficient

```r
# Model for dataset penguins
# fit's formula is flipper_length_mm ~ body_mass_g

# Intercept Interpretation
predict(fit, newdata =  data.frame(body_mass_g=0))

# Angular Coefficient Interpretation
# How the change in 1 unit of measure of x, translates
# in the unit of measure of y
predict(fit, newdata =  data.frame(body_mass_g=4001)) - predict(fit, newdata =  data.frame
(body_mass_g=4000))
```

7.6 Hypothesis Testing

The form below is appropriate when EST follows a gaussian-like distribution with a certain mean (typically the true value of the parameter) and standard deviation SE.

$$TS = \frac{EST - HYP}{SE}$$

- **TS**, the test statistic which (in this case) can be compared to a standard gaussian-like distribution. **If large we reject the null hypothesis**, since there is enough evidence against the hypothesized value for the parameter.
  - t_{obs} can be found in the `t value` column of the summary, while the relative p-value can be found in the `p value` column. When the p-value is smaller than the significance level, we can reject the null hypothesis.
- **EST**, the estimate for parameters of interest
  - It can be found in the `Estimate` column of the summary
- **HYP**, a hypothesized value of the parameter
  - The default value for R is 0
- **SE**, the standard error of the estimate
  - It can be found in the `Std. Error` column of the summary or in the corresponding cell of the `vcov()` matrix

Say we wish to test whether some β_j has a certain value β_j^*. The test hypothesis of interest is:

$$t_{obs} = \frac{\widehat{\beta}_j - \beta_j^*}{SE[\widehat{\beta}_j]} \sim t_{n-p} \begin{cases} H_0 : \beta_j = \beta_j^* \\ H_A : \beta_j \neq \beta_j^* \end{cases}$$

```r
# If we have only  the estimated values (est),
# and the vcov() matrix, we want to test it like this:
# we assume a significance level of 5%, alpha = 0.05
est + qt(c(0.025, 0.975), df = n-2) * sqrt(vcov[i,j])
# where vcov[i,j] is the estimated variance

# To check whether hyp=3 is in the interval
# First we get TS, say we have est=1.5
((1.5)-3)/sqrt(vcov[i,j])
# It should be outside the following range to
# reject the null hypothesis
qt(c(.025, .975), df = n-p)
```

By default R tests all beta parameters $H_0 : \beta_j = 0$ vs $H_A : \beta_j \neq 0$

- For the SLRM, if the null hypothesis $H_1 : \beta_1 = 0$ is true, we are basically considering a model where there is no significant linear relationship between X and Y.
- It can only detect straight line relationships. On the other hand the significance of the linear terms does not mean that the relationship is entirely linear.

8 Notation for MLR

The multiple linear regression extends the case of the SLR, translating from a 2 dimensional space to a p-dimensional space thanks to the additive nature of the model:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1(p-1)} \\ 1 & x_{21} & x_{22} & \cdots & x_{2(p-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n(p-1)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} = X\beta + \epsilon, \tag{1}$$

```r
fit <- lm(formula = target ~ predictor1 + predictor2 + ... , data = dataset,...)
```

- Y, *target* or *response* vector of size $1 \times n$. It is a random vector;
  - y, vector of **observed** (real) values for the response variable;
  - y_i, the i-th observed value;
  - $\widehat{Y}$, the estimated model
  - $\widehat{y} = X\widehat{\beta} = Hy$, vector of **estimated** values for the response variable;

    ```r
    y_hat <- X %*% solve(t(X) %*% X) %*% t(X) %*% y
    ```

  - $\widehat{y}_i$, the i-th estimated value;
  - $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, the sample mean
- $X \sim \mathcal{N}(\mu, \Sigma)$, *input* matrix or **model /design matrix** of size $n \times p$ where is the number of rows and p is the number of beta parameters. Remember that for this version $\mu_j = \mathbf{E}[X_j]$ and $\sigma_{ij} = Cov[X_i, X_j]$

    ```r
    # To create it
    X <- cbind(rep(1, n), dataset$predictor1, dataset$predictor2, ...)
    # To get it
    model.matrix(fit)
    ```

  - X^T, the transpose of the design matrix

    ```r
    t(X)
    ```

  - X_i, a predictor vector
  - $x_{i,j}$, an observed value
  - $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, the sample mean
- $H = (X^T X)^{-1} X^T$, the hortogonal projection matrix
- β the vector of the beta parameters
  - $\widehat{\beta} = (X^T X)^{-1} X^T Y$, the vector of beta parameters' estimates.

    ```r
    beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
    ```

- ε, the vector of the errors
  - $e = y - \hat{y}$ the vector of the residual value
- $s_e = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p}}$ the estimator of σ

```r
# from the model
summary(fit)$sigma
# manually
e <- y - y_hat
s_e <- sqrt(t(e) %*% e / (n - p))
# optionally
sqrt(sum((y - y_hat)    2) / (n - p))
```

9 Multiple Linear Regression - Multivariate Gaussian

9.1 Assumptions for MLR

The assumptions are the same as the ones we had before, but we should add some more in this case:

- We must highlight the fact that there is only a single parameter σ^2 sigma for the variance of the errors.
- We should add that this is only possible when the columns of X are **linearly independent**, otherwise we are not able to invert the matrix.
- X is said to have a $n - variate$ Gaussian Distribution $\mathcal{N}()$
- Y is conditionally normal

9.2 Interpretation

The interpretations changes now:

- We still minimize the MSE, but we need to compute p estimating equations. For ease of computation we exploit the Matrix representation.
- In fact, we are representing $\hat{y}$ as the transformation of the original observed values y (from a n-dimensional space), through the design matrix H (to a p-dimensional hyperplane). If the model is successful the structure in the data should be captured in those p dimensions, leaving just random variation in the residuals which lie in an $n - p$ dimensional space.
- The p-dimensional plane will pass by the **intercept** at $(\bar{x}_1, ..., \bar{x}_p, \bar{y})$
- Thanks to the **linear independence** of the predictors, each β_i **represents the change in** Y for two instances, given that all the other x_i (with $i \neq j$) have fixed values, which **only differ by one unit of measure** of x_i (which in the summary corresponds to the estimate). Their interpretation are strictly related to the unit of measure and range of the predictors they are related to.
  - For example, if $\hat{\beta}_1 = 0.005$ and the unit of measure of x_1 is in grams, then (given all other predictors have fixed values) the target estimate $\hat{y}$ will increase of $+0.005$ (target's) units when β_1 increases of 1 (unit of measure of the predictor x_1).
  - If $\hat{\beta}_1 > \hat{\beta}_2$ it does not mean that $\hat{\beta}_2$ is less "relevant". It is important to consider they `Std. Error` or s_e before dragging any conclusion.
  - Mind that later, when including interactions, the interpretation will change.
- The multiple linear regression model assumes that each predictor variable makes a separate contribution to the expected response, that these contributions add up without any interaction, and that each predictors contribution is linear.

10 Formulas for MLR

For ease of computing we now reason in terms of matrix and from the estimating equations we can derive the estimates for the beta parameters:

$$\widehat{\beta} = ((X^T X)^{-1} X^T y) \sim \mathcal{N}_p(\beta, \sigma^2 (X^T X)^{-1})$$

```r
beta_hat <- as.vector(solve(t(X) %*% X) %*% t(X) %*% y))
```

$$\widehat{\beta}_j \sim \mathcal{N}(\beta_j, \sigma^2 C_{jj})$$
$$\downarrow$$
$$t_{obs} = \frac{\widehat{\beta}_j - \beta_j}{SE[\hat{\beta}_j]} = \frac{\widehat{\beta}_j - \beta_j}{s_e \sqrt{C_{jj}}} \sim t_{n-p}$$

- $\mathbf{E}[\hat{\beta}] = \beta$ — $\mathbf{E}[\hat{\beta}_j] = \beta_j$ is an unbiased estimator
- $Var[\hat{\beta}] = \sigma^2 (X^T X)^{-1}$ — $Var[\hat{\beta}_j] = \sigma^2 C_{jj}$, where $C = (X^T X)^{-1}$

```r
    C <- solve(t(X) %*% X)
    s <- sqrt(sum((y - y_hat)^2) / (n - p))

```

- $SE[\hat{\beta}] = s_e \sqrt{\text{diag}(X^T X)^{-1}}$ — $SE[\hat{\beta}_j] = s_e \sqrt{C_{jj}}$

```r
    s_e <- s*sqrt(diag(C))
    # From the model this would be
    sqrt(diag(vcov(fit)))

```

We can estimate the coefficients in R either manually or through the usual way

```r
    n <- nrow(penguins)
    # a column with 1s
    X <- cbind(rep(1, n), penguins$body_mass_g, penguins$bill_length_mm)
    p <- ncol(X)
    y <- penguins$flipper_length_mm
    beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
    t(beta.hat)

    #we obtain the same through
    pfit <- lm(flipper_length_mm ~ body_mass_g + bill_length_mm, data = penguins)
    coef(pfit)
```

More over we can show the model matrix to check if we missed something

```r
(model.matrix(fit); (X)
```

We can then compute the estimations through:

$$\widehat{y} = X\widehat{\beta} = X(X^T X)^{-1} X^T y = Hy$$

For what concerns the variability we account for the error through s_e^2 which is an unbiased estimator of σ^2:

$$s_e^2 = \frac{\sum_{i=1}^n (y_i - \widehat{y}_i)^2}{n-p} = \frac{e^T e}{n-p}$$

11 Inference for MLR

11.1 Confidence Intervals for $\widehat{\beta}_j$

11.1.1 Parameters

We can construct confidence intervals for each of the $\widehat{\beta}_j$

$$est \pm CRIT \times SE$$
$$\downarrow$$
$$\widehat{\beta}_j \pm t_{\alpha/2, n-p} \cdot s_e \sqrt{C_{jj}}$$

```r
# Example
confint(fit, level = 0.99)
```

However, they tend to be overconfident.

11.1.2 Multivariate Parameters

We use the diagonal elements of $\hat{\beta}$ to construct confidence intervals for β_j

- On the hand we can estimate the covariance between the estimates through `vcov(fit)`
- It might be easier to look at the correlation `cov2cor(vcov(fit))`

Remember that if we are interested in confidence interval of **several parameters** we should take into account the correlation between the estimates.

11.1.3 Bivariate Parameters

We can plot an ellipse

- The ellipse displays a 95% joint confidence region for two regression coefficients
- The orientation reflects the correlation between the estimates. When they are not correlated the ellipse will tend to be circular.

11.2 Confidence Intervals for $\widehat{m}(x_0)$

Here we consider the uncertainty around the $\mathbb{E}[\hat{y}(x_0)]$, this is the typical value we can expect at some predictor value x_0

The estimate of $\mathbb{E}[Y|x = x]$ is given by $\hat{y}(x_0) = x_0^T\beta$

- it is unbiased (at this point there is no need to rewrite it)
- $SE[\hat{y}(x_0)] = s_e\sqrt{x_0^T(X^TX)^{-1}x_0}$

$$est \pm CRIT \times SE$$

$$\downarrow$$

$$\hat{y}_0 \pm t_{\alpha/2,n-p} \cdot s_e\sqrt{x_0^T(X^TX)^{-1}x_0}$$

In R we use `predict`

```r
new.peng <- data.frame(body_mass_g = c(5000, 5000), bill_length_mm = c(45, 35)); new.peng
     body_mass_g     bill_length_mm
1    5000            45
2    5000            35

(cint <- predict(pfit, newdata = new.peng, interval = "confidence", level = 0.99))
     fit          lwr          upr
1    211.8369     210.4808     213.1930
2    206.3966     203.5755     209.2176
```

If too wide, the interval might be suspect

We should always check whether the values are in the range of the observed ones, even through a plot. In fact, we might have added extreme values and this would be normal since we are extrapolating.

```r
range(penguins$body_mass_g)
[1] 2700 6300

range(penguins$bill_length_mm)
[1] 32.1 59.6
```

11.3 Prediction Intervals for new observations

Here we consider the uncertainty around $[\hat{y}|x = x_0]$ which is used to predict Y_o. As we always do, we need to consider that a new observation of y has more variance (that's why we have a 1+ under the square root) due to fact we use an estimator of σ^2.

- $SE[\hat{y}(x_0)] = s_e\sqrt{1 + x_0^T(X^TX)^{-1}x_0}$

$$est \pm CRIT \times SE$$
$$\downarrow$$
$$\hat{y}_0 \pm t_{\alpha/2,n-p} \cdot s_e\sqrt{1 + x_0^T(X^TX)^{-1}x_0}$$

11.4 Evaluating Uncertainty for predictions

11.5 Hypothesis Testing

The test always takes the same form

$$TS = \frac{EST - HYP}{SE}$$
$$\downarrow$$
$$\frac{\widehat{\beta}_j - \beta_j^*}{s_e\sqrt{C_{jj}}} \sim t_{n-p} \begin{cases} H_0 : \beta_j = \beta_j^* \\ H_A : \beta_j \neq \beta_j^* \end{cases}$$

Normally R considers the HYP to be 0, which is in the summary `summary(fit)$coef`. This test can interpreted as whether there is a relevant relationship between the predictor j, given that terms for other predictors are still present in the model.

12 Model Selection

Model selection is the process of analyzing what predictors should be included in a model in order to balance its complexity and adaptability. In statistics we want to have $n > p$ and **p should be ideally the smallest subset of features**, while we hope the observations to be $n \to \infty$.

12.1 Nested Models

Nested models are two models where one model $Y_{B,i}$ is nested inside another $Y_{A,i}$: $Y_{B,i}$ contains a **subset** of $\mathbf{q-1}$ predictors from **only** the larger model $Y_{A,i}$ which has $\mathbf{p-1}$ predictors.

- the inequality $q < p$ for beta-parameters implies $q - 1 < p - 1$ the same for the predictors.
- This definition can be extended to many nested models for each full model, since we can start from an empty model which only returns the mean and find all the possible combinations.

We consider a full general linear additive linear model with **p** beta-parameters:

$$Y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_{(q-1)} x_{i(q-1)} + ... + \beta_{(p-1)} x_{i(p-1)} + \varepsilon_i$$

We can proceed by comparing different subsets of predictors to achieve the best model for our goals through the null hypothesis testing:

- $H_0 : \beta q = \beta q + 1 = ... = \beta p - 1 = 0 \ \rightarrow \ \mathbf{Y_i} = \beta_\mathbf{0} + \beta_\mathbf{1} \mathbf{x_{i1}} + ... + \beta_{\mathbf{(q-1)}} \mathbf{x_{i(q-1)}} + \varepsilon_i$
  - The **null model**, with $q - 1$ predictors, where the predicted values are denoted as $\hat{y_{0,i}}$
  - Interpretation: "**None** of the predictors (from q+1 to p-1) show significant linear relationship with Y". We are trying to say that discarding those predictors, would let us achieve the same results with a simpler model.
- H_A : **At least one of** $\beta j \neq 0$, with $j = q, ..., p - 1$
  - The **full model**, with p-1 predictors, where the predicted values are denoted as $\hat{y_{A,i}}$
  - Interpretation: "**At least one** of the predictors (from q+1 to p-1) shows a significant linear relationship with Y". However, we have no clue which one it could be.

We need to remember these are extreme cases, as we usually want to consider models which are parsimonious but not as simple as the null model. So the process would be starting from a model with all the possible features (or the best ones from a domain knowledge research), and going through a continuous removal and comparison of the results obtained.

12.1.1 Significance of Regression

We can still make a use of the decomposition of variance. As a simpler model implies **less uncertainty**, we might choose it when the difference between the estimates of $SS_{RES}(H_0) - SS_{RES}(H_A)$ is small, since the fit of the smaller model is almost as good as the larger model and so we would prefer the smaller (more parsimonoius) model on the grounds of simplicity. On the other hand, if the difference is large, then the superior fit of the larger model would be preferred.

$$\frac{SS_{RES}(H_0) - SS_{RES}(H_A)}{SS_{RES}(H_A)} = \frac{\sum_{i=1}^{n} (\hat{y}_{A,i} - \hat{y}_{0,i})^2}{\sum_{i=1}^{n} (y_i - \hat{y}_{A,i})^2}$$

This suggests using this as a good test statistic.

12.1.2 R^2's problem

One way is to try and maximize the r-squared $R^2 \to 1$, but it might be misleading since it increases when the number of predictors increases.

12.1.3 Significance of Regression Test

The test follow these steps:

1. Formulate the two nested models to compare

```
# H0
null_pengs <- lm(flipper_length_mm    body_mass_g + bill_length_mm, data = penguins)

# HA
full_pengs <- lm(flipper_length_mm  body_mass_g + bill_length_mm+bill_depth_mm+year,
    data = penguins)
```

2. Decompose SS_{TOT} in SS_{RES} and SS_{REG} into an **AN**alysis **O**f **VA**riance table

Source	Sum of Squares	Degrees of Freedom	Mean Square	F
diff	$\sum_{i=1}^{n}(\hat{y}_{A,i} - \hat{y}_{0,i})^2$	$p - q$	$SS_{diff}/(p-q)$	MS_{diff}/MS_{res}
full	$\sum_{i=1}^{n}(y_i - \hat{y}_{A,i})^2$	$n - p$	$SS_{res}/(n-p)$	
null	$\sum_{i=1}^{n}(y_i - \hat{y}_{0,i})^2$	$n - q$		

```
# SSdiff
(ssdiff <- sum((fitted(full_pengs) - fitted(null_pengs)) ^2))
[1] 3543.322

# SSR (For Full)
sum(resid(full_pengs)^2)
[1] 10055.06

# SSR (For Null)
sum(resid(null_pengs)^2)
[1] 13598.38

# Degrees of Freedom: Diff
(dfdiff <- length(coef(full_pengs)) - length(coef(null_pengs)))
[1] 2

# Degrees of Freedom: Full
dffull <- length(resid(full_pengs)) - length(coef(full_pengs))
[1] 328

# Degrees of Freedom: Null
length(resid(null_pengs)) - length(coef(null_pengs))
[1] 330
```

3. Compute the F-statistic

   - $F = \frac{\sum_{i=1}^{n}(\hat{Y}_{A,i} - \bar{Y})^2/(p-q)}{\sum_{i=1}^{n}(Y_i - \hat{Y}_{A,i})^2/(n-p)} \sim \mathcal{F}(p-q, n-p)$
   - if it is LARGE, we can reject H_0 as the estimated y $\hat{y}_{A,i}$ are very different from $\bar{y}$ and the null model is not better. In fact, a large value of the statistic corresponds to a large portion of the variance being explained by the regression.

```
# F value
f_obs <- ((ssdiff/dfdiff) / (sum(resid(full_pengs)^ 2)/full_pengs$df.resid))
[1] 57.79227
```

4. Calculate the p-value $Pr(F > F_{obs})$ which should be small to reject the null hypothesis.

In R we can simply:

```
anova(null_pengs, full_pengs)

Analysis of Variance Table
Model 1: flipper_length_mm    body_mass_g + bill_length_mm
Model 2: flipper_length_mm    body_mass_g + bill_length_mm + bill_depth_mmyear

    Res.Df  RSS     Df  Sum of Sq   F       Pr(>F)
1   330     13598
2   328     10055   2   3543.3      57.792  < 2.2e-16 ***
```

These measures do not represent well the goodness of fit, nor the adaptability of the model. In fact, they do not say anything about the validity of the assumptions so if the assumption of data normality is not valid, the models built here are useless! This is something we want to be able to test when facing comparisons of different models.

12.2 Quality Criterion

A more general approach to model selection

- Includes ways to evaluate:
  - Goodness of fit, **no overfitting nor underfitting**
  - Adaptability to **new data**
  - Validity of **assumptions**
- Takes into account:
  - **Different** kinds of models to compare (not only nested ones)
  - **Size** of the model

12.2.1 Information Criteria - IC

12.2.2 Akaikes' Information Criterion - AIC

12.2.3 Bayesian Information Criterion - BIC

12.2.4 Adjusted R^2

12.3 Variable Selection

12.3.1 Forward Stepwise Selection

12.3.2 Backward Stepwise Selection

12.3.3 Stepwise Search

12.3.4 Inference after Selection

12.3.5 Validation-based selection

12.3.6 Leave-One-Out-Cross-Validation

13 Categorical Predictors and Interactions

13.1 Some Vocabulary First

13.2 Factors

Each one of the following encoding has its own interpretation, when fitted in a model. Furthermore the interactions, also change the meaning of regression.

13.2.1 Binary Encoding - 2 levels

13.2.2 Ordinal Encoding - 2+ levels

13.2.3 One-Hot-Encoding - 2+ levels

13.2.4 Dummy Encoding - 2+ levels

13.3 Interactions

13.4 Interpretations

14 Model Checking

A way to check the assumptions